Motion Graphs

Lucas Kovar
Michael Gleicher
University of Wisconsin – Madison

Frédéric Pighin
University of Southern California
Institute for Creative Technologies

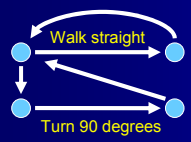
Directable Motion Capture

Motion capture *records* an event.

How can we improve control over motion capture without sacrificing quality?

Motion Graphs

Idea: automatically add transitions within a motion database



Edge = clip
Node = choice point
Walk = motion

Quality: restrict transitions

Control: build walks that meet constraints

Talk Outline

1. Related Work
2. Building Motion Graphs
3. Using Motion Graphs
4. Path Synthesis

Related Work: Multi-Target Blending

Use blends to create a parameterized space of motions

- Multilinear interpolation (Wiley and Hahn '97)
- Radial basis functions (Rose et al. '98)

Individual clips, not sequences of clips

Related Work: Statistical Models

State-based statistical models

- Bowden 2000
- Brand and Hertzmann 2000
- Galata et al. 2001
- Li et al. 2002

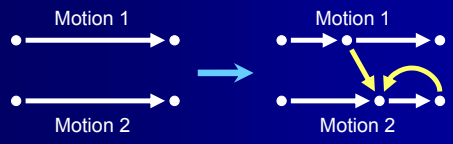
Related Work: Graph-Based Motion

Procedural Motion	Perlin '95
Physically based motion	Faloutsos et al. 2001
Image based motion	Schödl et al. 2000
Simplified Characters	Lamouret and van de Panne '96
Move trees	Mizuguchi et al. 2001
Simple keyframing	Molina-Tanco and Hilton 2000
Higher-level control	This session

Building Motion Graphs

Start with a database of motions, each with type and constraint information.

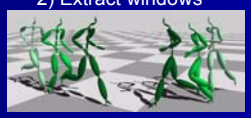
Goal: add transitions at opportune points.



Finding Transitions

1) Initial frames

2) Extract windows

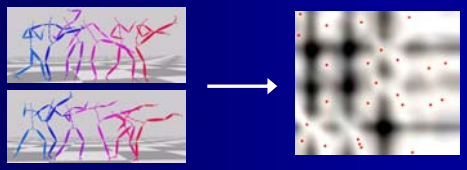


3) Convert to point clouds

4) Align point clouds and sum squared distances

Finding Transitions (cont.)

Every pair of frames now has a "distance".

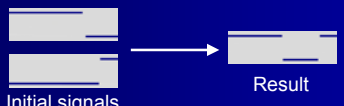


Extract the local minima below some threshold.
The threshold depends on the type of motion!

Creating Blends

Skeletal parameters are blended linearly.

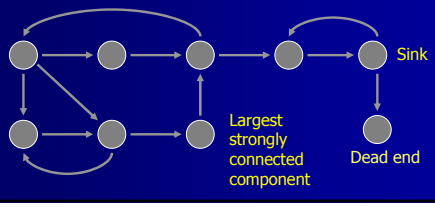
Constraints are blended as binary signals.



Pruning the Graph

Problem: sinks and dead ends.

Solution: extract the largest strongly connected component for each motion type.



Searching for Motion

Extracting motion typically requires a search.

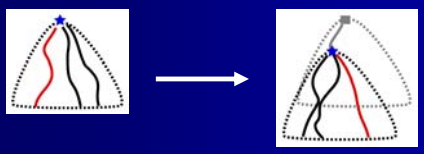
Function name	What it determines
Cost function (g)	Cost of adding an edge
Edge legality function (L)	Edges that can be added
Halting function (h)	Whether search can stop

Total cost: $\sum_{i=1}^{n} g([e_1,\ldots e_{i-1}], e_i)$

$g([e_1,\ldots e_{i-1}], e_i) \geq 0$

Searching For Motion (cont.)

The search is incremental. At each stage we consider all walks of n frames and keep k frames of the lowest-cost legal walk.

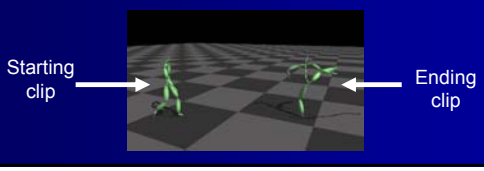


Branch and bound speeds the search.

Selecting Search Functions

A flawed setup: put two clips on the floor and find a walk that

1. Connects the clips
2. Positions and orients them as given

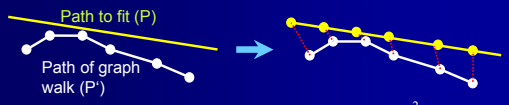


Selecting Search Functions cont.

1. Cost functions should guide the *entire* motion, not just certain parts.
2. Speed is as important as accuracy
3. Use the edge legality function to avoid tricky tradeoffs.

Path Synthesis

Goal: extract motion that travels along a path.



$$g([e_1,\ldots e_{i-1}], e_i) = \sum_{i=1}^{k} \left\| P'(s(e_i)) - P(s(e_i)) \right\|^2$$

Can require different styles of motion along different parts of the path.

Conclusion

Summary

- Build motion graphs automatically
- Search for particular motions
- Application to path synthesis

Thanks…

Motion capture data
• Demian Gordon (Digital Alchemist)
• House of Moves Studios
• Ohio State University ACCAD

Equipment and Software Donations
• Intel • IBM • Alias/Wavefront
• NVidia • Discreet • Pixar
• USC School of Film and Television

Financial Support
• NSF grants CCR-9984506 and IIS-0097456
• U.S. Army
• University of Wisconsin WARF